Single-pass constant- and variable-bit-rate MPEG-2 video compression

by N. Mohsenian
R. Rajagopalan
C. A. Gonzales

Most real-time MPEG-2 encoders are designed to perform in a constant-bit-rate (CBR) mode, in which buffer constraints are imposed to circumvent large deviations from a desired rate at any instant in time. Although such streams are generally good-quality sequences, certain types of operations or environments call for a more efficient real-time CBR encoder. The first part of the paper describes how a better-quality CBR video stream can be produced by estimating the relative complexity of a picture in comparison with the average complexity of the partially encoded stream and using it to adjust the compression parameters in a single-pass mode of operation. Our CBR encoder is particularly attractive for digital broadcast and editing environments, in which representations of higher-fidelity video objects in both display and freeze modes are constantly pursued. The second part of the paper describes the real-time generation of video streams with a variable-bit-rate (VBR) encoder. This mode of operation is highly desirable for home entertainment and recreational events. We propose a robust single-pass VBR video encoder algorithm which is capable of learning and adapting itself to the complexity of image segments and thereafter creating streams which have constant visual picture quality. The new VBR scheme displays a better performance than the CBR encoder, particularly when special effects such as scene transitions, fades, or luminance changes are to be compressed. Both CBR and VBR encoders are fully compliant with the MPEG-2 standard and are easily implementable with IBM encoder architecture. Compression results for the new single-pass encoding algorithms and comparisons with previous CBR schemes are provided. The result suggests the suitability of our VBR approach for record/playback in storage media such as digital video disc (DVD) players, disk-based camcorders, and digital videocassette recorders (DVCRs). It further reflects the importance of our single-pass CBR scheme for providers of broadcast services, for which it allows more video programs to be allocated to a selected communication link, and for in-studio applications, for which it greatly facilitates visual analysis of captured streams.

0018-8646/99/$5.00 © 1999 IBM

1. Introduction

The standardization of MPEG-2 [1] has greatly facilitated the transmission, representation, storage, and manipulation of digital video in various environments such as broadcast television, wireless communications, consumer electronics, and multimedia computers. Further, applications ranging from desktop publishing on personal computers (PCs) to home authoring with digital video disc (DVD) players demonstrate the major role played by the MPEG-2 standard in promoting new storage media for consumer video and interactive multimedia. This has enabled the home user to download video streams from a satellite system or an Internet site in order to create DVD video programs or multimedia presentations using a recordable medium. Although the syntax and specification of MPEG-2 bitstreams and multimedia programs (as in DVD titles) are well defined by the international standards, the actual encoding parameters and functions, which should lead to a fully compliant bitstream, have been and are the subject of many research efforts. The main challenge is how to achieve a close-to-optimum video quality in a compressed stream while reducing the amount of information in the source. This is because the statistical nature of any video source is either not known *a priori* or will change over time, and a true estimation of source distribution can be anywhere from computationally extensive, as in non-real-time multipass encoding, to almost impossible for real-time encoding.

The nonstationary nature of images makes them inherently variable; compression optimality for MPEG-2 coder–decoders (codecs) is achieved by carefully selecting a set of spatial or temporal image analyzers, quantizers, and variable-length entropy coders, of which some are frozen by the standard and some are to be defined by the designer. The results are variable[1] streams which require a sophisticated buffering scheme to smooth out the variability of the signal before transmission over a fixed bandwidth is carried out. The receiver will have a similar buffering policy to convert the fixed-channel rate to variable streams prior to the decoding and display of each picture.

Since design and development of an MPEG-2 video encoder can become cumbersome as a result of formulating several mathematically or perceptually derived parameters, typical approaches [2] enforce a constant bit rate (CBR) for a group of pictures (GOP) regardless of the complexity of the video interval. This scheme assumes equal weighting of bit distribution among GOPs and reduces the degree of freedom of the encoding task. In short, the problem is reduced to minimizing (maximizing) the GOP distortion (quality) subject to a constant target rate. By CBR we mean that the sustainable rate of the encoded video stream per GOP is close to a constant target rate, but the instantaneous rate changes per picture depending on picture type[2] or the quantization scaler. Another advantage of a CBR stream is that the transmitted signal may be terminated at any time and the user is assured of maintaining a rate close to the target rate. All CBR MPEG-2 encoders enforce different quantizing scalers for each picture type to achieve good-quality streams within a GOP. This method of compression works adequately when the complexity of the source varies slowly over time and therefore the encoding algorithm has time to adjust itself. However, if the statistical features of the source change rapidly over time, a constant-bit-rate operation may result in good picture quality for a short time window (e.g., a few frames or a GOP) and discontinuous quality when the whole video is perceived.

One way to improve the perceptual quality of a CBR stream while maintaining its constant rate from start to finish is to identify "difficult-to-encode" pictures and increase their bit budget accordingly. Conventional approaches to real-time CBR encoding use picture-to-picture[3] correlations in terms of complexity measures to predict the level of difficulty of a picture. In this paper we improve upon this first-order prediction by estimating the encoding difficulty of a picture on the basis of the complexity of all previously encoded pictures. Our single-pass CBR algorithm employs an infinite impulse response (IIR) filter to dynamically determine a nominal value which represents the degree of difficulty of the partially analyzed video stream in real time. We claim that the number of bits consumed by a more (or less) "difficult-to-encode" picture can be adjusted by comparing the encoding difficulty of the substream against a local measurement. This method of real-time compression improves the overall quality of the video and maintains a constant rate throughout the stream. Therefore, it can potentially become a key encoding element in cable television (CATV), direct satellite, and terrestrial broadcast arenas as well as mobile and asynchronous transfer mode (ATM) communications.

Since we argued that the video is inherently variable, an even better compressed stream can be created by employing a variable-bit-rate (VBR) encoder algorithm. Applications for such a scheme are plentiful. VBR can be used for networks that employ a dynamic bandwidth, as in ATMs, or it can be exploited as a means of achieving statistical multiplexing for digital broadcast satellites. Other major arenas which can benefit from the use of a VBR encoder are consumer video and interactive multimedia, where recording of high-quality pictures onto

[1] Here the term *variability* means the instantaneous rate of the video.

[2] The MPEG-2 standard uses I (intrapicture), P (predicted), and B (bidirectionally predicted) types.

[3] Pictures must be of the same type for an accurate prediction.

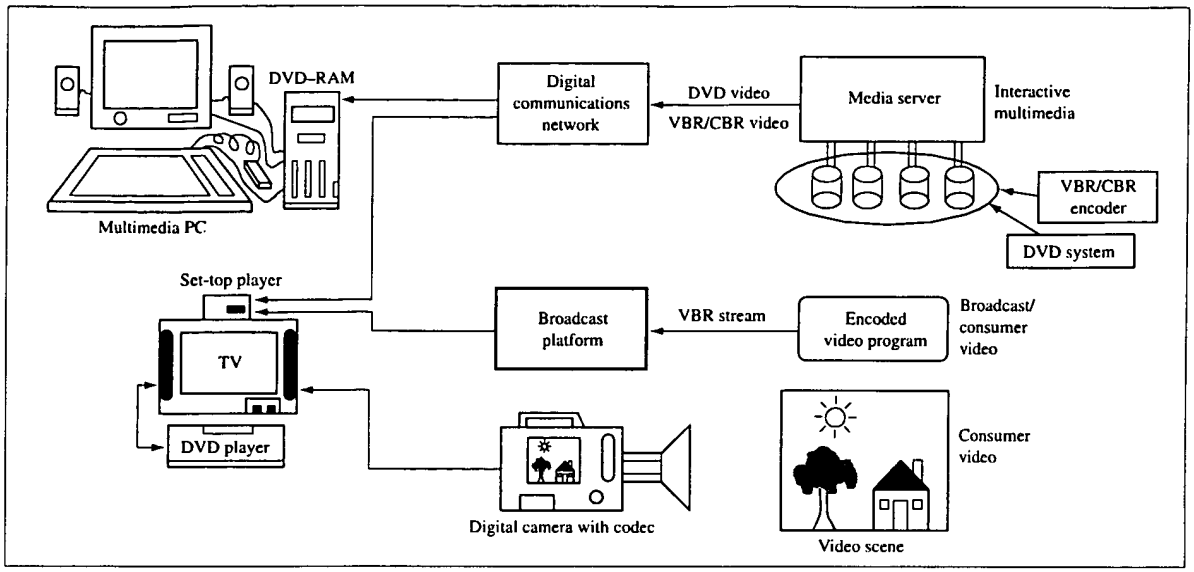

Figure 1

Single-pass real-time VBR encoding applications.

a storage medium is desired. For the aforementioned environments, DVD video movies are created to be played on set-top players, while DVD–RAM drives are used for multimedia productions in computers. Some examples of end-to-end solutions of the above industry segments are shown in **Figure 1**.

Some attempts at formulating a VBR video coding scheme have appeared in the literature. Most of them propose constraining the quantizer-scale parameter to a user-defined target value for long periods of time [3]. This value is adjusted via network negotiation or monitoring the system buffer to match quality of service (QoS). Although simple intuition suggests that fixing the quantizer scaler would redistribute the amount of bits among GOPs of differing complexity, there are no guarantees of obtaining a constant video quality. Further, in ATM applications an encoded stream is usually queued before a bandwidth is available in the network. This lead time enables pre-encoding tasks to be performed on particular streams subject to the overall channel bandwidth and buffer fullness. Others have investigated a simple variable-bandwidth estimation model for the number of ATM cells generated through packetization of video sources [4]. In this allocation scheme, a fixed quantizer-scale parameter is again used by the encoder. Authors in [5] have presented a VBR video encoder which takes advantage of the limits of the human observer to improve the perceived quality of the decoded sequence while maintaining the output bit rate within permitted bounds.

For consumer video, a popular mode of encoding operation has been the multipass VBR scheme. For example, high-end DVD mastering can be accomplished with an encoding system which typically uses a three-step procedure. The first step encodes the video source in a CBR mode and gathers a set of predefined statistical features. This information is then used to compute a set of optimized quantization parameters which closely match the source distribution of the video data and would provide a better compressed stream during a second-pass encoding. The last step is a postprocessing task which creates the final DVD format. The intermediate procedure can be carried out several times to produce an ideal video program. This type of application has the advantage that many lookahead parameters are known *a priori* when the VBR video coding algorithm is implemented. Unfortunately, for applications such as home DVD productions, writable DVDs for PC, digital camcorders, and DVCRs, the availability of a platform capable of analyzing and performing a multipass VBR encoding is unrealistic. Therefore, a VBR scheme is desired which is superior to CBR and can be implemented in one pass. This paper is intended to introduce such an algorithm.

In the first part of the paper we propose a new single-pass CBR encoder algorithm which is tailor-made for real-time compression and can easily be realized with the IBM

encoder architecture. We add a further level of complexity to the CBR encoder and introduce a new real-time single-pass VBR encoder in the second part of the paper. Our VBR scheme employs a causal predictive model to distinguish the "hardness" or "softness" of the incoming video material *on the fly* and adapts itself accordingly. Moreover, it relies on a perceptual model to improve the quality of the stressful segments of the stream and produce high-fidelity video. The perceptual model is also responsible for adjustment of the average rate of the stream. The rest of the paper is organized as follows. Section 2 gives some background information for CBR and VBR algorithms. The new CBR encoding technique is presented in Section 3, and the VBR rate-control algorithm is defined in Section 4. Simulation results are given in Section 5, and concluding remarks are provided in Section 6.

2. Background formulation

An MPEG-2 sequence is typically partitioned into small intervals called GOPs (groups of pictures), which in turn are categorized by picture types I (intracoded or intrapicture), P (predicted), and B (bidirectionally predicted) [1]. The number of bits per GOP is distributed such that the allocation for an I-picture is more than that for a P-picture. This is because a P-picture uses a motion-estimation (ME) technique to estimate its content; as a result, a motion-compensated frame difference (MCFD) with a lower entropy than the original source is encoded. B-pictures use the smallest number of bits because their ME techniques are more intensive than those for P-pictures. This method provides a basis for maintaining the same picture quality within a GOP when pictures of different types are encoded. We may further lower the bit allocation of B-pictures, since they will not be used to estimate other pictures. Each picture type is subdivided into square blocks of pixels called macroblocks. Since producing an efficient compressed stream is the main mission of an MPEG-2 encoder, devising a robust rate-control (RC) algorithm becomes an integral part of the encoding task. The rate-control algorithm monitors the number of bits that should be allocated to each picture or macroblock on the basis of type or image feature, respectively. Moreover, it should ensure that the decoder buffer does not experience an overflow or underflow during the time the stream is received from the communication channel and prepared for decoding. In short, there is a need for the following items in any type of MPEG-2 compression scheme:

- Target bits (or quantization scalers) for picture types.
- Buffer regulation to avoid overflow/underflow conditions.
- Maintenance of a target rate or consumption of no more than the bit budget.
- A rate-control strategy which ensures that all of the above are monitored/satisfied.

In the remainder of this section, we build a framework for a CBR compression scheme, since most of these ideas are needed for the new single-pass CBR and VBR approaches. We assume that pictures of a video sequence can be modeled as a memoryless Gaussian source with variance σ^2 varying from picture to picture; hence, their rate-distortion (R-D) relationship is defined by $R(D) = (1/2) \log (\sigma^2/D)$. Experimental results in the literature suggest that a similar behavior exists between the rate R of the source and the quantization factor Q [6],

$$R(Q) = b_1 \log \frac{b_2}{Q}. \tag{1}$$

Instead of using the logarithmic model of (1), we adopt a simpler hyperbolic relationship which is easily implementable in real time and has proved to be an effective realization of the R-Q model [2]. The simplified equation takes the form

$$R(Q) = \frac{\chi}{Q}. \tag{2}$$

Equation (2) indicates that the picture rate is inversely proportional to the quantization factor. χ is a predefined measure of complexity for each picture type. However, since the nature of the source changes over time, a new complexity measure is required prior to encoding of each picture type. This parameter is usually computed using the past encoding parameters, e.g., bits, quantization factor, and/or some lookahead statistics. For each GOP[4] of the MPEG-2 stream, we enforce a total number of bits given by C_ℓ,

$$\sum_x N^x R^x = C_\ell \qquad x = I, P, B. \tag{3}$$

Index ℓ denotes the GOP number, and x is the picture type. N^x is the number of pictures of type x in a GOP, and R^x is the number of target bits for picture type x. For a CBR sequence we have $C_\ell = C_{\text{gop}}$, where C_{gop} is a fixed number of GOP bits. For a given C_ℓ, the video quality of the GOP is maximized by minimizing the average sum of the quantization scalers subject to the condition of Equation (3),

$$\Psi = \frac{\sum_x N^x Q^x}{\sum_x N^x}. \tag{4}$$

[4] For simplicity, we have assumed a fixed GOP structure throughout this paper, but an adaptive GOP structure may also be employed in our MPEG-2 framework.

Instead of minimizing Ψ subject to the constraint of (3), we remove this condition and use the Lagrange multiplier λ to minimize the Lagrangian cost Y:

$$Y = \Psi - \lambda C_\ell . \tag{5}$$

With the aid of the rate-quantization model described in (2), the target for each picture type is deduced:

$$R^x = \frac{\chi^x C_\ell}{\sum_x \chi^x N^x} . \tag{6}$$

Targets in Equation (6) represent only ideal picture bits; the actual bits would almost always deviate from this. The accumulated error must be computed and fed back to the rate-control algorithm to ensure that the final MPEG-2 bitstream meets the average bit rate or the total bit budget. Let $C_{\ell,\text{ideal}}$ and $C_{\ell,\text{actual}}$ represent the ideal and actual bits for GOP ℓ, respectively, and let $\delta_{\ell,\text{gop}} = C_{\ell,\text{actual}} - C_{\ell,\text{ideal}}$ be the difference between the two. Further, let $R_{i,\text{ideal}}$ and $R_{i,\text{actual}}$ represent the ideal and actual bits for picture i, respectively, and $\delta_{i,\text{pic}} = R_{i,\text{actual}} - R_{i,\text{ideal}}$ be the difference between the two. After n pictures have been encoded, the total accumulated error can be computed as

$$\sum_{\ell=0}^{n_g-1} \delta_{\ell,\text{gop}} + \sum_{i=0}^{n-n_gG-1} \delta_{i,\text{pic}} = \Delta_{n-1,\text{gop}} + \Delta_{n-1,\text{pic}} . \tag{7}$$

The size of the GOP is given by $G = \Sigma_x N^x$, and $n_g = \lfloor n/G \rfloor$ is the number of fully encoded GOPs. Suberror accumulation for all processed GOPs is given by $\Delta_{n-1,\text{gop}}$, while $\Delta_{n-1,\text{pic}}$ is the suberror accumulation for the last but not yet finished GOP in the encoding order. The ideal picture target can now be adjusted for overproduction or underproduction of bits, computed from previously encoded pictures. The new set of ideal bits prior to encoding picture n belonging to GOP ($\ell = n_g$) is

$$R^x_{n,\text{ideal}} = \frac{\chi^x_{n-1}(C_\ell - \alpha_1\Delta_{n-1,\text{gop}} - \alpha_2\Delta_{n-1,\text{pic}})}{\sum_x \chi^x_{n-1} N^x} , \tag{8}$$

where α_1 and α_2 are constants that indicate how aggressively this adjustment is carried out. For a CBR scheme it is recommended to use $\alpha_1 = \alpha_2 = \alpha$ with $\Delta_{n-1} = \Delta_{n-1,\text{gop}} + \Delta_{n-1,\text{pic}}$, and compute the adjusted picture bits as

$$R^x_{n,\text{ideal}} = \frac{\chi^x_{n-1}(C_{\text{gop}} - \alpha\Delta_{n-1})}{\sum_x \chi^x_{n-1} N^x} . \tag{9}$$

The rate-quantization framework described so far is based on the assumptions that the decoder buffer is of infinite size and that a large enough number of bits is always available. However, this is not true in a real-life scenario, where the buffer size is limited and defined

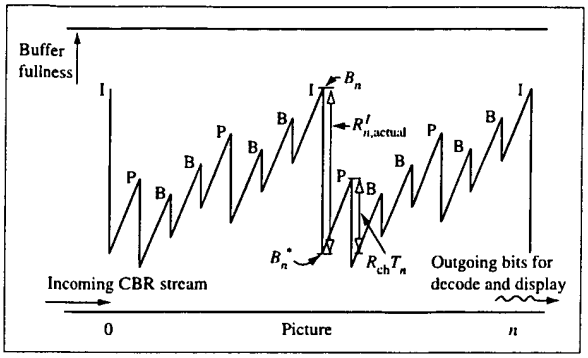

Figure 2

Picture bit fluctuations for a decoder video buffer verifier.

by the MPEG-2 standard [1]. The encoding scheme is responsible for eliminating any overflow or underflow condition that the decoder buffer may encounter. This is accomplished by examining a hypothetical decoder buffer, i.e., video buffer verifier (VBV), and computing lower/upper bounds on the number of bits assigned for a picture type. The lower bound should be a large enough nonnegative number to prevent the overflow condition, while the upper bound should not be larger than a predetermined value above which the VBV buffer will underflow. Let B_n and B^*_n be the decoder buffer fullness before and after picture n is removed, respectively. B^*_n is then computed as

$$B^*_n = B_n - R_{n,\text{actual}} , \tag{10}$$

and the buffer fullness before removing the next picture is

$$B_{n+1} = B^*_n + R_{ch}T_n , \tag{11}$$

where R_{ch} is the channel rate (in Mb/s) at which the decoder buffer is being filled, and T_n is the display period for picture n. **Figure 2** shows how the occupancy of a VBV buffer changes over time. Before we remove picture n, we have all the information in the buffer available to us; therefore, the upper bound U_n becomes the buffer occupancy. To compute the lower bound L_n, suppose $R_{n,\text{actual}}$ is just that, and picture n has been removed at once and delivered for display. The decoder buffer is filled at the rate R_{ch} during this period, and the VBV buffer fullness before removing picture ($n + 1$) would have to be smaller than the total size of the video buffer verifier, i.e., B_{vbv}, in order to prevent overflow. Therefore, the nominal values by which the picture target bits are bounded are given by

$$U_n = B_n,$$

$$L_n = \max(0, B_n + R_{ch}T_n - B_{vbv}). \tag{12}$$

After each picture is encoded, the complexity measures χ^x are updated, and a new target based on Equation (8) is computed for the next picture. This target should meet the constraints imposed by (12) to satisfy the VBV buffer policy. We clip the ideal picture bits using the picture bounds U_n and L_n:

$$R_{n,\text{ideal}} = \begin{cases} U_n & \text{if } (R_{n,\text{ideal}} > U_n), \\ L_n & \text{else if } (R_{n,\text{ideal}} < L_n), \\ R_{n,\text{ideal}} & \text{else.} \end{cases} \tag{13}$$

Finally, a quantization scaler Q, which is defined on the hyperbola of Equation (2), is obtained. It should be noted that each picture type has its own composite R-Q curve, and further, for each picture type the Q factor may be adjusted slightly to ensure that all pictures are perceived to be of equal quality. In this paper we do not define how to incorporate perceptual effects, via modulating the Q factor, for a macroblock-level rate control during the encoding task; the reader can refer to [2] for this procedure. However, we must describe a strategy which ensures that the ideal target number of bits for each picture type is met. The above condition can be satisfied by monitoring the actual number of bits computed for a set of encoded macroblocks against the ideal average number of bits of a macroblock. Let $r_{m,n,\text{actual}}$ represent the actual number of bits calculated for macroblock m in picture n. In order to track how closely we match the ideal picture bit, a parameter $\Delta_{m,n,\text{mb}}$ is defined:

$$\Delta_{m,n,\text{mb}} = \sum_{p=0}^{m-1} r_{p,n,\text{actual}} - \frac{mR_{n,\text{ideal}}}{M_{\text{mb}}}, \tag{14}$$

where M_{mb} is the total number of macroblocks in a picture. A positive $\Delta_{m,n,\text{mb}}$ indicates an overproduction of bits at the macroblock level; therefore, the picture quantizer must be increased for the next macroblock to satisfy the target number of bits. Similarly, a negative value dictates the opposite scenario. A macroblock-level rate-control strategy can be formulated to modulate the Q factor on the basis of overproduction or underproduction of bits. We accomplish this by defining a factor $q_{m,n}$, used to scale macroblock m in picture n, through [7]

$$q_{m,n} = Q_n \begin{cases} A^c(D_n, \Delta_{m,n,\text{mb}}) & \text{if } (\Delta_{m,n,\text{mb}} > 0), \\ A_f(E_n, \Delta_{m,n,\text{mb}}) & \text{else,} \end{cases} \tag{15}$$

where $A^c(.,.)$ and $A_f(.,.)$ are empirically derived functions which help in maintaining the picture target bits. They should behave such that $A^c(.,.)$ is always larger than 1 and $A_f(.,.)$ is smaller than or equal to 1. Q_n is the picture quantizer, and differential targets D_n and E_n are

$$D_n = U_n - R_{n,\text{ideal}},$$
$$E_n = R_{n,\text{ideal}} - L_n. \tag{16}$$

We have now defined a CBR MPEG-2 encoder specified by $\{R_i, Q_i\}_{i=0}^{n}$. In the following section, we describe how this framework can be modified to produce a better CBR stream.

3. Single-pass CBR video

Previous approaches to CBR video compression such as the schemes defined in Section 2 and in [2] have used only the encoding parameters of one previously encoded picture to estimate the R-Q relationship of a picture having the same type. After a picture is encoded, its complexity χ^x is updated as defined in [2] and used for the next picture of the same type,

$$\chi^x_{n-1} = \begin{cases} R^x_{n-1,\text{actual}} M_{\text{mb}}^{-1} \sum_{m=0}^{M_{\text{mb}}-1} q^x_{m,n-1} & \text{if } (n-1) \text{ is of type } x, \\ \chi^x_{n-2} & \text{for all other types,} \end{cases} \tag{17}$$

and the complexity measures are used to predict a quantization value for picture n,

$$Q^x_n = \frac{\chi^I_{n-1} N^I + \chi^P_{n-1} N^P + \chi^B_{n-1} N^B}{(C_{\text{gop}} - \alpha\Delta_{n-1})}. \tag{18}$$

An examination of Equation (18) reveals that a first-order prediction is used to estimate the position of an (R_i, Q_i) pair for each picture type to be encoded. This method of estimation can be improved by observing the long-term complexity of all pictures processed so far and determining the relative complexity of a current picture to be encoded. The long-term complexity $\bar{\chi}^x_{n-1}$ of a picture type x is defined as

$$\bar{\chi}^x_{n-1} = \begin{cases} \dfrac{(\Gamma_x - 1)\bar{\chi}^x_{n-2} + \chi^x_{n-1}}{\Gamma_x} & \text{if } (n-1) \text{ is of type } x, \\ \bar{\chi}^x_{n-2} & \text{for all other types.} \end{cases} \tag{19}$$

Equation (19) characterizes an IIR filter structure with the complexity of a picture as an input and the average complexity of a subset of the stream as the output. Coefficient Γ_x is fixed over time for a picture type x and reflects how strongly the output of the filter is dependent on the previous input and output samples. The canonical structure of the IIR filter represents an efficient and simple way to respond quickly to the statistical variations of a video sequence in real time. The recurrence in Equation (19) can be factored to write the average complexity in terms of the individual complexity of encoded pictures,

$$\bar{X}^x_{n-1} = \frac{(\Gamma_x - 1)^{n-1}}{\Gamma_x^{n-1}} \bar{X}^x_0 + \sum_{i=1}^{n-1} \frac{(\Gamma_x - 1)^{n-(1+i)}}{\Gamma_x^{n-i}} X^x_i , \quad (20)$$

with

$$\bar{X}^I_0 = \bar{X}_{\text{init}} ,$$

$$\bar{X}^x_0 = \kappa_x \bar{X}^I_0 \qquad \text{for } x \in \{P, B\}. \quad (21)$$

Since each GOP typically starts with an I-picture, the selection of the magnitude of $\bar{X}_{\text{init}}$ determines how well we may encode the first few Is of the sequence. Constants κ_P and κ_B are intended to maintain a complexity ratio among the start-up values. We now take advantage of the fact that before compression of a "difficult-to-encode" picture, the differential target D_n may be a large positive number which can be used to adjust the ideal target allocation of such a picture. Similarly, the bit allocation of an "easy-to-encode" picture can be lowered on the basis of the differential target E_n. It should be noted that the notion of difficulty of encoding reflects the relative complexity of a picture among all processed pictures. This conception is substituted for the expression "more (less) difficult to encode" throughout this formulation. In order to identify the level of difficulty of a picture type, we compare its complexity against the output of the IIR filter, which indicates the history of all previously encoded pictures of this type. If $(X^x_{n-1} \geq \bar{X}^x_{n-1})$, we label this picture as "difficult to encode" and increase its bit allocation. For the opposite scenario, i.e., $(X^x_{n-1} < \bar{X}^x_{n-1})$, the picture is labeled as "easy to encode," and bit allocation is lowered. The actual adjustments are defined by modifying $R_{n,\text{ideal}}$ of Equation (13) through

$$R^x_{n,\text{ideal}} = \begin{cases} \min\left(R^x_{n,\text{ideal}} + \dfrac{X^x_{n-1} - \bar{X}^x_{n-1}}{X^x_{n-1} + \bar{X}^x_{n-1}} \rho_{\max} D_n,\ \nu_{\max} R^x_{n,\text{ideal}} \right) \\ \quad \text{if } (X^x_{n-1} \geq \bar{X}^x_{n-1}), \\ \max\left(R^x_{n,\text{ideal}} + \dfrac{X^x_{n-1} - \bar{X}^x_{n-1}}{X^x_{n-1} + \bar{X}^x_{n-1}} \rho_{\min} E_n,\ \nu_{\min} R^x_{n,\text{ideal}} \right) \\ \quad \text{else}, \end{cases} \quad (22)$$

and the picture quantizer is computed as

$$Q^x_n = \frac{X^x_{n-1}}{R^x_{n,\text{ideal}}} . \quad (23)$$

The new targets in Equation (22) suggest that for extreme cases, i.e., pictures that are very difficult (or easy) to encode, we may come very close to the upper or lower bound of a picture and force the decoder buffer to underflow or overflow. To resolve this problem, constants $\rho_{\max}$ and $\rho_{\min}$ are incorporated to use only a portion of the available buffers D_n or E_n. The min and max operations, along with user-defined $\nu_{\max}$ and $\nu_{\min}$ parameters, are used to prevent a picture from using too many or too few bits. A further precaution is taken to avoid decoder buffer overflow by adding a guard band g_L to increase the lower picture bound:

$$L_{n,\text{adj}} = L_n + g_L ; \quad (24)$$

the new adjusted differential targets (buffers) are given by

$$D_{n,\text{adj}} = U_n - R_{n,\text{ideal}} - g_U ,$$

$$E_{n,\text{adj}} = R_{n,\text{ideal}} - L_{n,\text{adj}} . \quad (25)$$

Again, a guard band g_U is used to prevent the decoder buffer from underflowing. After the new target for a picture is set through the conditions given in Equation (22), we must still clip the bits using the adjusted lower bound $L_{n,\text{adj}}$ and the upper bound U_n. The updated parameters $D_{n,\text{adj}}$ and $E_{n,\text{adj}}$ are used to ensure that the new targets are achieved by implementing a macroblock-level rate-control strategy, as defined in (15). **Figure 3** displays a graphical representation of how a transformation of a plot of the relative complexity of a picture is used to set a new bit target. It further reflects how VBV buffer compliance is guaranteed and picture bit budgets are satisfied by enforcing the actual bit production to operate within certain limits and cling to the ideal bit allocation.

The robustness of any type of constant-bit-rate control algorithm is directly dependent on the speed with which it can respond to the changes of the video content of an incoming stream. For real-time applications in which an efficient hardware implementation must be realized with dedicated integrated circuits, the veracity of the RC algorithm may be severely tested under stressful conditions. This is because such customized solutions do not use any preprocessing functions to pre-analyze the nature of the stream; consequently, the availability of the true encoding parameters of a picture always lags the assigned values. A real-time RC algorithm must process and memorize a few GOPs of the same complexity before it can reach its optimality. We argue, on the basis of the following observations, that our new approach to CBR video compression, as formulated in Equation (22), offers a quicker response and better results than the conventional method of CBR encoding.

Consider an encoder that has processed a few "difficult-to-encode" pictures and, further, that this new group of pictures belong to the same GOP. The increase in picture difficulty may be due to the sudden (or gradual) appearance of a high degree of spatial image detail, an increase in the velocity of many moving objects of different scales, directional changes of objects, or some form of higher-order combinations. At the start of a new GOP, we must encode an I-picture. Since we have already compressed one difficult I-picture, there is a high probability that the next I-picture is also "difficult to encode," as reflected by the complexity of the previously analyzed picture. Our CBR scheme can adjust to this

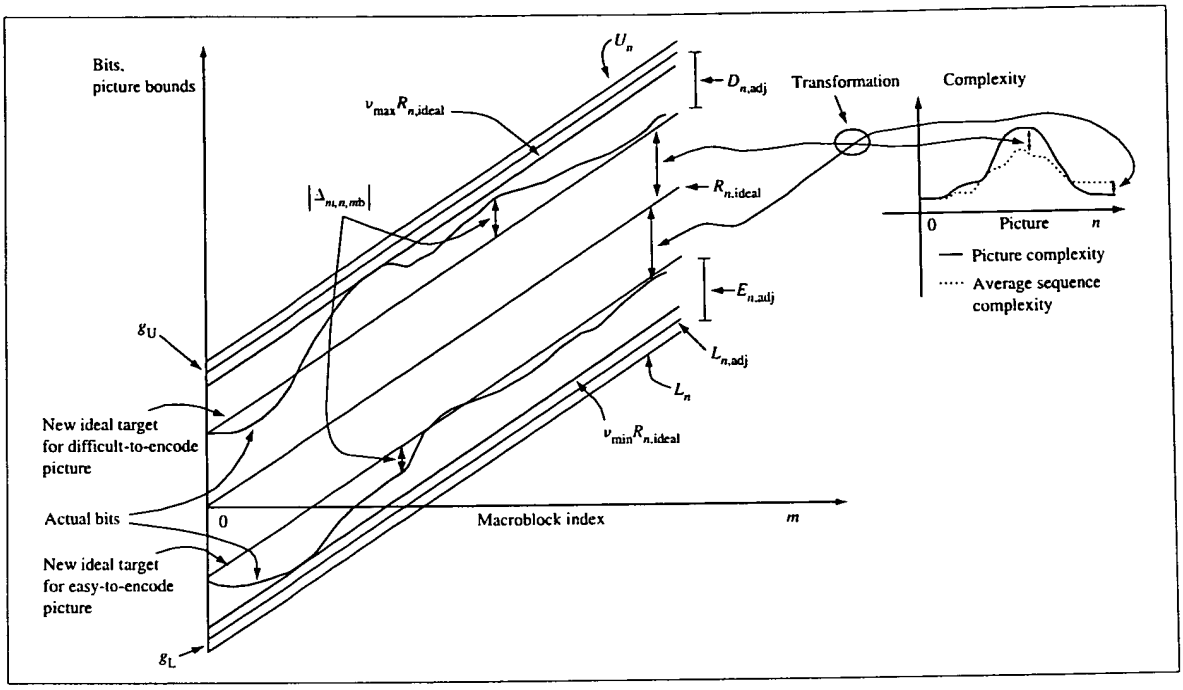

Figure 3

Target adjustments for "difficult/easy-to-encode" pictures in the new single-pass CBR scheme.

picture much more quickly by knowing that the I-picture is significantly different from the rest of the video sequence. This observation is made by comparing the output of the IIR filter against the estimated I-complexity. As a result, the I-picture consumes more bits than the normal bit allocation generated by a conventional method of CBR encoding. Therefore, a higher-quality I-picture is reconstructed at the decoder output. I-pictures are used as references to predict a block of pixels in P- and B-pictures. A better prediction is now obtained for the non-intracoded pictures, resulting in better reconstruction of such pictures at the cost of a small number of bits. Hence, we improve the perceptual quality of a GOP while still adhering to the constant bit rate of the stream. If an easy picture is to be encoded, it consumes fewer bits, and the remaining bit budget is used to encode the future pictures of a GOP. Overall, the number of bits and the picture quality average out over an "easy-to-encode" GOP. It should be noted that the human observer finds image distortions in "difficult-to-encode" pictures most annoying; for long video programs, a small degradation in "easy-to-encode" pictures is tolerated. Further assessment of the argument presented in this section and comparison with the method described in Section 2 are provided in Section 5, which discusses the simulation results.

4. VBR video

The CBR rate-control strategy of the previous section is inspired by the fact that for a fixed target C_{gop}, constant quality is achieved within a GOP by modulating the quantization parameters. Any statistical variations or bit offshoots are exploited to help stabilize the CBR RC algorithm over time and maintain a desired rate. In addition, an MPEG-2 CBR encoder takes advantage of a set of universal constants and predetermined initial complexity measures to ensure a certain ratio between the number of bits allocated among different picture types [2]. However, efforts in classifying a group of continuous pictures, such as GOPs or video segments, into different types of time intervals in terms of complexity "hardness" or "softness" have been limited for real-time single-pass MPEG-2 encoding. We define "hardness" ("softness") of the video by the large (small) number of bits that it requires to produce high-fidelity results. For multipass CBR or VBR encoding, such information is known; hence, quality improvements can be made. One way of achieving single-pass VBR compression is to compare the Q factor, derived by the CBR algorithm, against a fixed parameter [8]:

$$Q_{vbr} = \max(Q_{fix}, Q). \tag{26}$$

The method in (26) is intended to provide an upper bound on the quality of pictures which belong to soft video segments. When soft segments are analyzed, it is very likely that we have ($Q_{fix} > Q$). Therefore, a quantization scaler, larger than the values normally assigned by the CBR encoder, is assigned to pictures in soft segments. A VBR stream is produced by distributing the surplus bits among the hard segments of the video. Q_{fix} may be obtained through experimentation with a large number of sequences, but finding a near-optimum value is difficult if not impossible. A better scheme can be formulated by calculating a Q_{fix} scaler in real time using prior image statistics [7]. The concept behind the method in [7] is to combine the Q_{fix} approach of [8] with that of the CBR RC algorithm.

Our real-time single-pass VBR encoder exploits an R-Q compression model to differentiate the degree of "hardness" or "softness" of video segments, each segment corresponding to a particular hyperbola similar to the one defined by (2). The actual encoding parameters of the video segments are computed along this hyperbola. We also use an R-Q perceptual model to prioritize the video segments in terms of visual importance. To satisfy the average rate of the VBR stream, the position of the perceptual model must be changed over time. In this paper we specify two methods for meeting this condition. The perceptual model and the VBR RC algorithms are described next.

- *Rate-quantization perceptual model*

The VBR scheme proposed in this paper is conceptually motivated by the fact that each video segment is associated with a level of encoding difficulty, and this difficulty can be measured by various source statistics or compression parameters such as total picture bits, quantization scaler, spatial activity, temporal activity, signal-to-noise ratio, or any combination of them. A larger number of bits should be allocated to a video segment with a high level of encoding difficulty. This is a different approach from that of the CBR RC algorithm, in which a fixed number of bits is allocated to each GOP regardless of the degree of complexity of the source. Research efforts have shown that for a large number of test cases composed of complex, moderate, and easy materials, a strong correlation exists between the rate of the video interval and its corresponding quantization scaler [9]:

$$R_a \propto (a_1 + a_2 Q_a^{\beta}). \tag{27}$$

The R-Q relationship of (27) is deduced on the basis of the criterion that all types of video intervals should be perceived equally. Further, it suggests that difficult video segments producing a large quantization scaler should consume more than the average bit rate of the compressed stream, while easy segments would use smaller bits. This provides a natural building block in formulating a robust single-pass VBR encoding algorithm, since it takes advantage of the variability of the video source. The actual number of bits allocated to each interval is determined by the slope K of the perceptual model,

$$R_a = K(a_1 + a_2 Q_a^{\beta}) = KF(Q_a). \tag{28}$$

Constant K (measured in Mb/s) is modulated for each video interval to ensure that the average rate of the compressed stream meets the desired target. For the case of $\beta = 1.0$ the perceptual model reduces to a linear relationship.

- *VBR rate-control algorithms*

The efficiency of a single-pass VBR encoder is assessed by the speed with which its rate-control algorithm can learn and adjust itself to the "softness" or "hardness" of the video stream. For regions where image discontinuities or special effects occur, degradations in picture quality should be minimized. Since for single-pass encoding, image statistics are limited by the previously analyzed pictures, the learning rate of the RC algorithm must be adequate to predict the content of the future video intervals, yet not aggressive enough to result in algorithmic instabilities. One way to solve the twofold problem is to adjust the quality of the encoded stream for every time interval and let the RC algorithm learn the local content of each picture within that time interval.

In this subsection we claim that a VBR video stream S_{vbr} is a concatenation of several contiguous video intervals, each operating at a different CBR bit rate. We further use the GOP terminology as the definition of a video (or time) interval for the remainder of the paper, and compute the quantization parameters of the single-pass VBR algorithm for each GOP. Therefore, S_{vbr} can be partitioned into a number of piecewise-continuous GOPs specified by $\{S_t\}_{t=0}^{m}$. For GOP S_t we define an average number of bits $\bar{C}_t$ and an average quantization scaler $\bar{Q}_t$ such that $[\bar{C}_t \bar{Q}_t] = G^{-1}[C_t Q_t]$. We modify the terms of (28) to form a dependency between the average number of bits and quantization scaler of GOP S_t,

$$\bar{C}_t = f^{-1} KF(\bar{Q}_t). \tag{29}$$

The picture rate of the video is defined by f. Equation (29) represents the perceptual rate-quantization model for a GOP. We further take advantage of the theoretical rate-quantization model and assume a hyperbolic dependency between the average number of bits and quantization scaler of a GOP as in (2). The GOP rate-quantization model takes the form of

$$\bar{C}_t(\bar{Q}_t) = \frac{X_t^c}{\bar{Q}_t}. \tag{30}$$

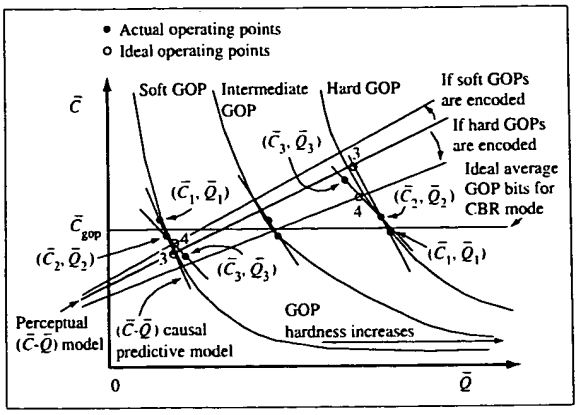

Figure 4

Diagram of the $\bar{C}$-$\bar{Q}$ models and the operating points of VBR rate-control algorithm 1.

The average complexity of a GOP is given by X_t^c. To understand how the two perceptual and theoretical rate-quantization models work in harmony, we should look at their graphical representations in **Figure 4**. The perceptual model is overlaid on the ideal rate-quantization behavior of GOPs. Once the VBR encoder processes a few GOPs, the $\bar{C}$-$\bar{Q}$ relationship of Equation (30) can be realized. Depending on the actual compression parameters, the model of Equation (30) characterizes a "hard" or "soft" GOP. Then the $\bar{C}$-$\bar{Q}$ perceptual model is used to emphasize the visual importance of a "hard" GOP. The positive slope of this model ensures that "hard" GOPs are assigned more bits. A more detailed description of the VBR rate-control algorithm is presented later in this section.

Since our goal is to formulate a VBR algorithm which is readily applicable for hardware implementations, we form a linear approximation for the model of Equation (30). The new model uses previously processed GOPs to construct a straight line for the $\bar{C}$-$\bar{Q}$ relationship. We call this line the $\bar{C}$-$\bar{Q}$ predictive model and define its slope and $\bar{C}$ – *intercept* by ξ_t and η_t, respectively. Next, we explain how to derive this model.

Let $(\bar{C}_{t-2,\text{actual}}, \bar{Q}_{t-2,\text{actual}})$ and $(\bar{C}_{t-1,\text{actual}}, \bar{Q}_{t-1,\text{actual}})$ represent the average number of bits and average quantization scaler pairs of GOPs S_{t-2} and S_{t-1} just encoded.[5] Then, the $\bar{C}$-$\bar{Q}$ relationship for GOP S_t to be encoded is given by

$$\bar{C}_t = -\xi_t \bar{Q}_t + \eta_t , \tag{31}$$

with

$$\begin{cases} \xi_t = \Delta\bar{C}_t(\Delta\bar{Q}_t)^{-1} & \eta_t = \bar{C}_{t-1,\text{actual}} + \xi_t\bar{Q}_{t-1,\text{actual}} , \\ \Delta\bar{C}_t = \bar{C}_{t-1,\text{actual}} - \bar{C}_{t-2,\text{actual}} & \Delta\bar{Q}_t = \bar{Q}_{t-2,\text{actual}} - \bar{Q}_{t-1,\text{actual}} . \end{cases}$$

Equation (31) defines the instantaneous rate-quantization behavior of a particular GOP under analysis and will change over time. To find the optimum operating point, we solve the causal predictive model of (31) together with the perceptual model of (29) and encode the next GOP at the average number of bits of

$$\bar{C}_t = \frac{K(\xi_t a_1 + \eta_t a_2)}{f\xi_t + a_2 K} . \tag{32}$$

Constant K is modulated for each GOP to ensure that the total number of bits produced by the VBR stream is not more than the size of the storage or retrieval device, e.g., a DVD disc. Assume that the total number of bits available is R_{TOT}. Then, after every GOP of the video sequence is analyzed and encoded, we can use the perceptual model of (29) to compute K:

$$K = \frac{fR_{\text{TOT}}}{G\sum_{t=0}^{N_{\text{gop}}-1} F(\bar{Q}_{t,\text{actual}})} . \tag{33}$$

The number of GOPs in the video sequence is given by N_{gop}. Given a number of pictures and a bit budget R_{TOT}, the single-pass VBR encoder has the responsibility of fitting all of the produced bits into the digital medium. To prevent overruns or underruns, R_{TOT} must be dynamically modified for each GOP. The adjusted budget $R_{t,\text{tot}}$ is obtained by subtracting the actual bits from R_{TOT} and is used to set a new slope,

$$K_t = \frac{fR_{t,\text{tot}}}{G\sum_{t=0}^{N_{\text{gop}}-1} F(\bar{Q}_{t,\text{actual}})} . \tag{34}$$

We call the VBR encoder which uses this method of bit assignment (or slope modulation) VBR method 1 (VBR-1). The denominator of Equation (34) can easily be computed in a multipass encoding scheme, but is not available for a single-pass real-time compression. Instead, we use a pre-encoded phantom sequence with a set of GOP quantizers defined by $\{\bar{Q}_t^*\}_{t=0}^{N_{\text{gop}}^*-1}$, and adjust the summation of Equation (34) after each GOP of the test sequence is encoded. N_{gop}^* is the number of encoded GOPs in the phantom sequence. The learning procedure for (34) is formulated as follows.

Learning procedure 1

1. Let P be a pre-encoded phantom defined by the tuple $\{N_{\text{gop}}^*, \{\bar{Q}_t^*\}_{t=0}^{N_{\text{gop}}^*-1}\}$, and set $Z_0 = [\Sigma_{t=0}^{N_{\text{gop}}^*-1} F(\bar{Q}_t^*)]$, $N_0 = N_{\text{gop}}^*$.

[5] A CBR or VBR MPEG-2 encoder can be used here.

2. Initialize the VBR-1 RC algorithm by encoding the first two GOPs of the video sequence, i.e., S_0 and S_1, at a rate of $(f\bar{C}_{gop})$ using a CBR MPEG-2 encoder. The nominal value of $(f\bar{C}_{gop})$ should correspond to the average rate of the VBR stream.
3. After $S_{\ell-1}$ is encoded, update Z_ℓ and N_ℓ as $Z_\ell = Z_{\ell-1} + \gamma F(\bar{Q}_{\ell-1,\text{actual}})$, $N_\ell = N_{\ell-1} + \gamma$, where γ is the update speed of the learning algorithm. Compute $C_{\ell-1,\text{actual}}$ and then the sum $\Sigma_{j=0}^{\ell-1} C_{j,\text{actual}}$.
   a. If $(\ell - 1 = 0) \Rightarrow$ increment ℓ by one, go to step 3.
   b. Otherwise, compute ξ_ℓ and η_ℓ according to the causal predictive model defined in (31).
4. Use the sum term of step 3 to determine the remaining bits $R_{\ell,\text{tot}}$ in the budget. Update the number of remaining pictures P_ℓ to be encoded.
5. Measure the new slope: $K_\ell = fR_{\ell,\text{tot}} N_\ell (P_\ell Z_\ell)^{-1}$.
6. Calculate the new target $\bar{C}_\ell$ and encode S_ℓ in VBR mode.
7. If there are unfinished GOPs in the sequence, go to step 3; otherwise stop.

Term Z_ℓ is intended to adapt itself to the image content of each GOP and smooth out the volatility of the VBR video. It acts as a safety measure to prohibit unrealistic bit allocations to GOPs of very high (low) complexity. Finally, the average number of bits of the GOP S_ℓ to be encoded by the VBR-1 encoder is set by

$$\bar{C}_\ell = \left(\frac{R_{\ell,\text{tot}} N_\ell}{P_\ell Z_\ell}\right)\left[\frac{\xi_\ell a_1 + \eta_\ell a_2}{\xi_\ell + a_2\left(\frac{R_{\ell,\text{tot}} N_\ell}{P_\ell Z_\ell}\right)}\right]. \tag{35}$$

To adjust the target bits of each picture type, $\bar{C}_\ell$ is multiplied by the GOP size G and Equation (8) is used to set the VBR picture bits. For a VBR scenario, upper and lower picture bounds are defined differently from those for the CBR mode of operation. An overflow condition cannot occur for the decoder buffer of a VBR codec. This is because the task of filling the decoder buffer is immediately stopped after the VBV buffer occupancy reaches its maximum level. Therefore, a lower bound of zero is imposed for picture bound. Moreover, the decoder buffer is filled at the maximum rate of R_{max},[6] set by the user. **Figure 5** shows an example of how the occupancy of a VBV buffer changes over time for the VBR mode of compression. We clip the target bits of Equation (8) by the newly derived picture bounds

$$U_n = \min\,(B_{vbv}, B_{n-1} + R_{max} T_{n-1} - R_{n-1,\text{actual}}),$$
$$L_n = 0. \tag{36}$$

Picture quantization scalers are computed as in (23) and the macroblock-level rate-control strategy of Section 2

[6] For DVD application, an R_{max} = 9.8 Mb/s is suggested.

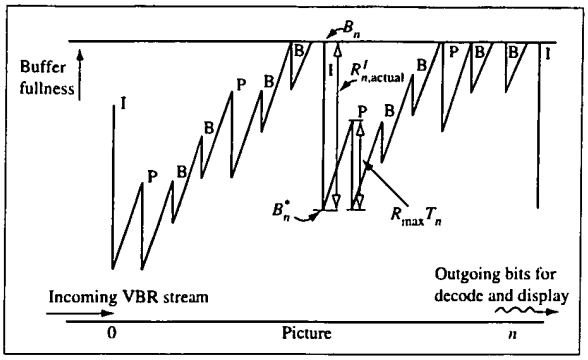

Figure 5

Picture-bit fluctuations for a decoder video buffer verifier in VBR mode of operation.

is used to maintain the picture targets. We have now defined a single-pass VBR MPEG-2 encoder specified by $\{R_{TOT}, \{R_i, Q_i\}_{i=0}^{n}\}$. The encoder operates along constellations which are formed by jointly solving for a time-varying perceptual model and a bank of $\bar{C}$-$\bar{Q}$ models. The relative ideal position of the $(\bar{C}_\ell, \bar{Q}_\ell)$ pair of a GOP within the constellation is first determined by the "softness" or "hardness" of the GOP and then adjusted by the remaining number of bits in the bit budget. The local position of the (R_i, Q_i) pair of a picture is represented by the R-Q model of the picture type. Figure 4 displays how a constellation of $(\bar{C}_\ell, \bar{Q}_\ell)$ pairs is formed by the VBR-1 encoder. In this figure, the constant line defined by $\bar{C} = \bar{C}_{gop}$ indicates the ideal location of all $(\bar{C}_\ell, \bar{Q}_\ell)$ pairs if they were to be encoded in CBR mode. Moreover, it reflects that the average quantization scalers increase as the hardness of GOPs increases. The ideal and actual operating points are depicted by light and dark circles, respectively. Let $(\bar{C}_1, \bar{Q}_1)$ and $(\bar{C}_2, \bar{Q}_2)$ be the first two pairs of a "soft" GOP, which are obtained by initialization of the VBR-1 encoder in the beginning of a video sequence. The next ideal operating point, denoted by "3," is the intersection of the perceptual model (solid line) and the causal predictive model. The position of point "3" indicates that an average number of bits smaller than the sequence average $\bar{C}_{gop}$ is allocated. However, the output of the encoder, i.e., point $(\bar{C}_3, \bar{Q}_3)$, will be different, and as a result, a new perceptual model (shown by the dotted line) is derived to meet the bit budget constraint. The pairs $(\bar{C}_2, \bar{Q}_2)$ and $(\bar{C}_3, \bar{Q}_3)$ are then used to obtain a new predictive model (shown by the dotted line) and compute a new operating point, i.e., "4." If the video sequence contains a large number of contiguous "soft" GOPs, the perceptual model will eventually converge to a line which

intersects the $\bar{C} = \bar{C}_{gop}$ line in close proximity to the CBR output pairs $(\bar{C}_1, \bar{Q}_1)$ and $(\bar{C}_2, \bar{Q}_2)$.

If the video material starts with "hard" GOPs, the location of the actual point $(\bar{C}_3, \bar{Q}_3)$ is above the $\bar{C} = \bar{C}_{gop}$ line. This ensures the optimization of the available bit budget for regions where a CBR encoder is most vulnerable, i.e., "hard" GOPs. In this scenario, a new perceptual model (shown by the dashed line) is formed to gradually lower the allocated GOP bits and comply with the bit budget constraint. This model, along with the new causal predictive model (also shown by the dashed line), determines the next operating point, denoted by "4." For the case in which the incoming sequence is composed only of "hard" GOPs, the perceptual model will eventually conform to a line which meets the $\bar{C} = \bar{C}_{gop}$ line at a location close to the $(\bar{C}_1, \bar{Q}_1)$ and $(\bar{C}_2, \bar{Q}_2)$ points. For a typical video program, it is unlikely that we operate along the same GOP (or a constellation of previously computed points) for a long duration of time. It is, however, likely that we jump out of a GOP to the next neighboring GOP after a short time. Therefore, we can deduce the following behavior. Before the perceptual model settles in a situation where it can monotonously take away from "hardness" or "softness" of a GOP, the RC algorithm will migrate to a new GOP. The actual quantization scaler values for previously encoded picture types determine the migration to a "harder" or "softer" GOP. For a future "harder" ("softer") GOP, we move to the right (left) of the previously encoded GOP as depicted in Figure 4. Hence, the point at which we intersect the perceptual model produces a number of GOP bits which is higher (lower) than the number of actual bits consumed by a previous GOP. This mechanism ensures that, for a collection of contiguous GOPs of similar "level of encoding difficulty," the effect of the "hardness" (or "softness") reduction or augmentation is distributed over the encoded content of the corresponding video interval, while the most (least) complex GOP still gets the largest (smallest) bit allocation relative to its lookalike GOPs.

While the VBR-1 encoding framework is based on modulating the slope K of the perceptual model, an alternative VBR RC algorithm can be formulated by translation of the perceptual model to meet the total number of bits set by the user. This method is labeled as VBR-2 and is defined next. For the VBR-2 bit-modulation scheme, we fix the slope K at a CBR rate of $f\bar{C}_{gop}$ and translate the position of the $\bar{C}$-$\bar{Q}$ perceptual model by solving for parameter a_1 of (29) through

$$a_1 = \frac{R_{TOT} - C_{gop} a_2 \sum_{\ell=0}^{N_{gop}-1} \bar{Q}_\ell}{C_{gop} N_{gop}}. \quad (37)$$

The sum term in the above equation is undefined for the real-time single-pass VBR-2 encoder. We use a pre-encoded phantom sequence (as in the previous algorithm) to initialize the VBR-2 RC algorithm. To ensure that the digital storage medium does not experience overruns/underruns, we monitor the remaining bits periodically and use them as the instantaneous total bit budget, i.e., $R_{\ell,tot}$. Translation of the perceptual model is varied as

$$a_1^\ell = \frac{P_\ell^{-1} R_{\ell,tot}}{\bar{C}_{gop}} - a_2 \frac{Y_\ell}{N_\ell} = r_a^\ell - a_2 E_\ell. \quad (38)$$

Term r_a^ℓ represents a ratio between the long-term rate of the GOPs not yet encoded and the CBR rate of the video stream. For "hard" GOPs we have $(\bar{C}_\ell > \bar{C}_{gop})$ and, therefore, ratio r_a^ℓ must become smaller over time to satisfy the total bit budget. Further enforcement is provided by E_ℓ to move the position of the $(\bar{C}_\ell, \bar{Q}_\ell)$ pairs downstream, along the constellation, to lower the rate of the compressed stream. The opposite scenario takes place if several soft GOPs are encoded over time. The average number of bits of each GOP is set as

$$\bar{C}_\ell = \frac{\xi_\ell a_1^\ell + \eta_\ell a_2}{\xi_\ell \bar{C}_{gop}^{-1} + a_2}. \quad (39)$$

Term Y_ℓ is initialized and updated during a learning procedure given below. The rest of the encoding parameters are defined with the formulation of the VBR-1 algorithm.

Learning procedure 2

1. Let P be a pre-encoded phantom defined by the tuple $\{N_{gop}^*, \{\bar{Q}_\ell^*\}_{\ell=0}^{N_{gop}^*-1}\}$, and set $Y_0 = (\Sigma_{\ell=0}^{N_{gop}^*-1} \bar{Q}_\ell^*)$, $N_0 = N_{gop}^*$.
2. Initialize the VBR-2 RC algorithm by encoding the first two GOPs of the video sequence, i.e., S_0 and S_1, at a rate of $(f\bar{C}_{gop})$ using a CBR MPEG-2 encoder. The nominal value of $(f\bar{C}_{gop})$ should correspond to the average rate of the VBR stream.
3. After $S_{\ell-1}$ is encoded, update Y_ℓ and N_ℓ as $Y_\ell = Y_{\ell-1} + \gamma \bar{Q}_{\ell-1,actual}$, $N_\ell = N_{\ell-1} + \gamma$, where γ is the update speed of the learning algorithm. Compute $C_{\ell-1,actual}$ and then the sum $\Sigma_{j=0}^{\ell-1} C_{j,actual}$.
   a. If $(\ell - 1 = 0) \Rightarrow$ increment ℓ by one, go to step 3.
   b. Otherwise, compute ξ_ℓ and η_ℓ according to the causal predictive model defined in (31).
4. Use the sum term of step 3 to determine the remaining bits $R_{\ell,tot}$ in the budget. Update the number of remaining pictures P_ℓ to be encoded.
5. Compute r_a^ℓ, E_ℓ, and the new translation factor a_1^ℓ.
6. Calculate the new target $\bar{C}_\ell$ and encode S_ℓ in VBR mode.
7. If there are unfinished GOPs in the sequence, go to step 3; otherwise stop.

Constraints on the VBV buffer occupancy and target adjustments are handled as before. **Figure 6** displays how the instantaneous causal predictive models are formed within each GOP and used in conjunction with a time-varying perceptual model to estimate a new $(\bar{C}_\ell, \bar{Q}_\ell)$ point for the VBR-2 encoder. If several soft (or hard) GOPs are encoded over time, the perceptual model will be shifted upward (or downward) to meet the bit budget constraint. In Section 5 we evaluate the efficiency and robustness of the single-pass VBR compression schemes by encoding several video sequences.

• *Scene transitions*

The reliability of any type of real-time single-pass RC algorithm is directly related to the number of pictures processed after the encoder start-up. The encoder is initialized with a set of statistical information which is updated over time as the encoder learns and adjusts itself to the spatial or temporal perturbation of image details. As a result, the output stream may suffer some degradations at the beginning of the video sequence. This distortion is often ignored, since the human visual system (HVS) requires a recovery time to comprehend changes in the image distortion [10], and, more significantly, no past encoded pictures are available as a reference at the beginning of the video. After a few GOPs are encoded, the learning curve of the RC algorithm reaches an equilibrium level and produces streams which display acceptable image quality upon decoding.

During the course of the encoding procedure, the RC algorithm can become unreliable. This is usually caused by special effects or naturally occurring phenomena such as scene cuts, slow- or fast-moving fades, and luminance changes. For such cases, the encoder scheme undergoes temporal transitions for which parameter adjustments become nearly impossible. These image discontinuities typically result in serious image degradations which the human observer finds very distracting. One way to overcome this problem is to detect the temporal position of the special effect in the video and replace the quantization parameters with a properly adjusted set. The former is easily derivable for hard scene cuts but can become complicated for fades, while the latter requires a new framework for deriving a new set of target rates and quantization scalers.

The existence of image discontinuities is even more troublesome for a VBR video because of the volatile nature of the compressed video parameters. Since our intention is to formulate a VBR RC algorithm offering continuous picture quality, we seek a VBR encoding scheme which displays graceful degradations when special effects are compressed. We accomplish this by detecting scenarios in which, as a result of scene transitions, the causal predictive model cannot provide a faithful

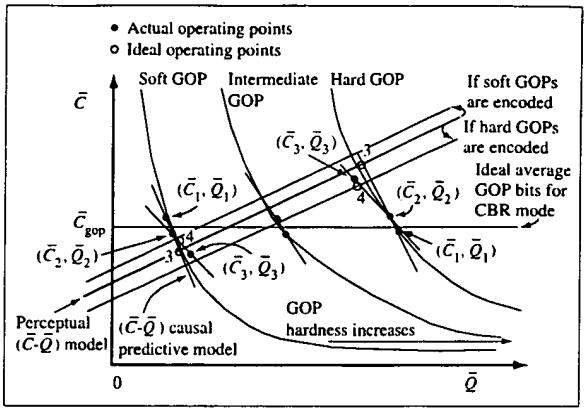

Figure 6

Diagram of the $\bar{C}$-$\bar{Q}$ models and the operating points of VBR rate-control algorithm 2.

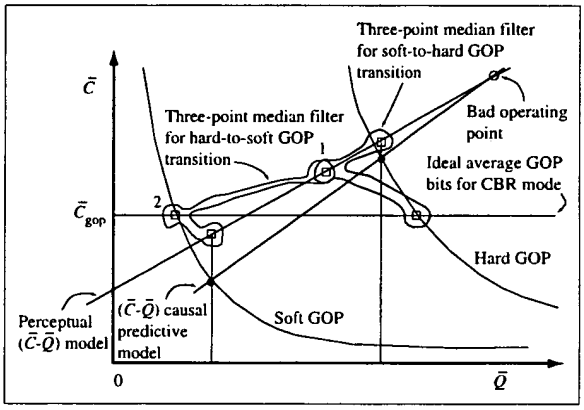

Figure 7

GOP bit adjustment during a scene transition using a median filtering technique.

estimation. Under these circumstances we set the GOP targets along a trajectory different from the procedure described in the subsection on VBR rate-control algorithms. **Figure 7** shows two cases (for the VBR-1 video encoder) in which transitions from "soft" GOP to "hard" GOP and vice versa make incorrect estimations for a future GOP to be encoded. The incorrect estimation is denoted by "bad operating point" in the figure. A much better estimation can be made by applying a three-point median filter to a set of carefully selected parameters. For the VBR-1 RC scheme, the new targets are derived using the following rule:

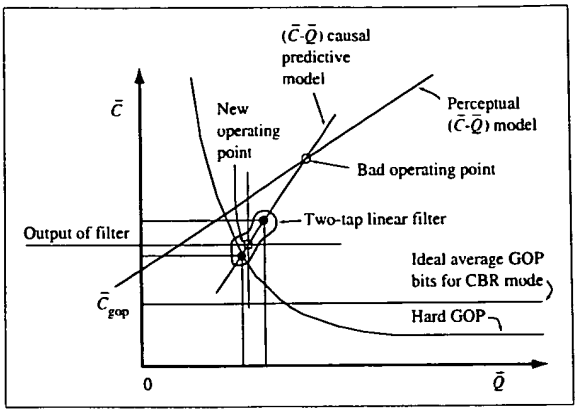

Figure 8

GOP bit adjustments for rate-control irregularities.

Table 1 Classes of the test sequence library.

Class	*Composition*
A	Consists mostly of different subsequences with high spatial detail and medium-to-high amounts of movement or vice versa
B	Special effects: luminance changes, zoom, rotation, etc.
C	Fade
D	Rapid scene changes
E	Equal amounts of difficult and easy video material

$$C_{\ell,\text{new}} = G\begin{cases}\text{median}\,[\mu_1 f^{-1}K_\ell F(\bar{Q}_{\ell-1,\text{actual}}),\ \mu_2\bar{C}_{\text{gop}},\ \mu_3 f^{-1}K_\ell] \\ \qquad \text{if } (\xi_\ell < 0), \\ \bar{C}_\ell \qquad \text{else;}\end{cases} \tag{40}$$

for the VBR-2 RC scheme, the targets are

$$C_{\ell,\text{new}} = G\begin{cases}\text{median}\left[\mu_4\bar{C}_{\text{gop}}(a_1' + a_2\bar{Q}_{\ell-1,\text{actual}}),\ \mu_5\bar{C}_{\text{gop}},\ \mu_6\dfrac{R_{\ell,\text{tot}}}{P_\ell}\right] \\ \qquad \text{if } (\xi_\ell < 0), \\ \bar{C}_\ell \qquad \text{else.}\end{cases} \tag{41}$$

The median filter is used to compensate for instability conditions that occur in the transition of video scenery among all types of GOPs, i.e., "soft" to "hard" and vice versa. The constant set $\{\mu_i | 1 \le i \le 6\}$ enforces a bound on the number of bits produced during a scene transition. For the "soft"-to-"hard" GOP transition, the output of the filter in Equation (40) is marked by "1" in Figure 7, while point "2" is the new estimate for the reverse transition. The nonlinear nature of the filtering schemes in (40) and (41) is an effective way of controlling the rate of the VBR encoder for higher-order temporal changes. Further, it does not require a local (picture-level) scene-change detector. However, if the encoder is equipped with a scene-change detector, we may take advantage of the GOP target allocation of Equations (40) and (41).

- *$\bar{C}$-$\bar{Q}$ irregularities*

The R-Q relationship of Equation (2) is known to work fairly well for a diverse class of video sequences and is a fundamental concept used for many real-time MPEG and non-MPEG compression systems. In this paper we have used it to build a causal predictive model to estimate the average bits of a present GOP to be encoded on the basis of the actual average number of GOP bits and average quantization factor of previous GOPs. However, this predictive model can become unreliable for some video segments even when there are no special effects, scene transitions, or scene cuts. The cause of this unreliability is related to the nonstationary nature of video sources, e.g., several highly detailed objects moving slowly across a background of significant luminance or chrominance image details, the presence of unwanted noise in the scene, etc. **Figure 8** displays two points derived by an encoder, with one being outside the composite $\bar{C}$-$\bar{Q}$ curve. This point is a result of the aforementioned criteria and does not obey the $\bar{C}$-$\bar{Q}$ relationship. Further, it will contribute to the calculation of a "bad operating point." Our VBR encoder suppresses this irregularity by making an additional contribution to the strategies defined in (40) and (41). We set the GOP bits according to the condition

$$C_{\ell,\text{new}} = \begin{cases}\max\,(C_{\ell,\text{new}},\ G\phi_{\max}\bar{C}_{\ell,\text{fil}}) \\ \quad \text{if } (\xi_\ell < 0) \wedge (\theta_L < |\Delta\bar{Q}| < \theta_U) \wedge (\bar{C}_{\ell,\text{fil}} > \bar{C}_{\text{gop}}), \\ \min\,(C_{\ell,\text{new}},\ G\phi_{\min}\bar{C}_{\ell,\text{fil}}) \\ \quad \text{else if } (\xi_\ell < 0) \wedge (\theta_L < |\Delta\bar{Q}| < \theta_U) \wedge (\bar{C}_{\ell,\text{fil}} < \bar{C}_{\text{gop}}), \\ G\bar{C}_{\text{gop}} \\ \quad \text{else if } (\xi_\ell < 0) \wedge (\theta_L < |\Delta\bar{Q}| < \theta_U) \wedge (\bar{C}_{\ell,\text{fil}} = \bar{C}_{\text{gop}}),\end{cases} \tag{42}$$

with

$$\bar{C}_{\ell,\text{fil}} = \frac{\omega_1\bar{C}_{\ell-2,\text{actual}} + \omega_2\bar{C}_{\ell-1,\text{actual}}}{\omega_1 + \omega_2}. \tag{43}$$

$\bar{C}_{\ell,\text{fil}}$ is an average number of GOP bits which is determined by applying a linear filter on the average number of bits of the previously encoded GOPs. The output of this filter is the new operating point, and (ω_1, ω_2) represent the weights of the filter. Constants $\phi_{\max}$ and $\phi_{\min}$ are used to scale the filter output. Thresholds θ_L and θ_U are chosen

Table 2 Average PSNRs and actual bit rates for all test sequences.

Sequence	*Type/Duration/Class*	*Old CBR*	*New CBR*	*VBR-1*	*VBR-2*
mpeg	NTSC/30 s/A	31.47 dB @ 4.0 Mb/s	31.64 dB @ 4.0 Mb/s	32.14 dB @ 4.0 Mb/s	32.10 dB @ 4.04 Mb/s
mixb	NTSC/30 s/A	37.01 dB @ 4.0 Mb/s	37.04 dB @ 3.99 Mb/s	37.02 dB @ 3.8 Mb/s	37.10 dB @ 3.96 Mb/s
LaBk	PAL/20 s/B	37.54 dB @ 4.0 Mb/s	37.48 dB @ 3.98 Mb/s	38.73 dB @ 4.0 Mb/s	38.52 dB @ 4.0 Mb/s
Sprite	NTSC/4 s/C	38.79 dB @ 4.0 Mb/s	38.73 dB @ 3.95 Mb/s	38.55 dB @ 3.85 Mb/s	38.58 dB @ 3.93 Mb/s
BkSz	NTSC/10 s/E	35.24 dB @ 4.0 Mb/s	35.28 dB @ 3.97 Mb/s	35.54 dB @ 3.64 Mb/s	35.62 dB @ 3.88 Mb/s
SzBk	NTSC/10 s/E	35.29 dB @ 3.98 Mb/s	35.37 dB @ 3.99 Mb/s	36.65 dB @ 4.02 Mb/s	36.51 dB @ 4.02 Mb/s
GreySep	NTSC/30 s/A, D	30.96 dB @ 4.01 Mb/s	31.22 dB @ 4.01 Mb/s	32.65 dB @ 4.02 Mb/s	32.66 dB @ 4.07 Mb/s
Singer	NTSC/4 s/B	31.85 dB @ 4.0 Mb/s	31.86 dB @ 4.03 Mb/s	31.96 dB @ 4.06 Mb/s	31.98 dB @ 4.05 Mb/s

under the assumption that a future GOP belongs to the same video segment as the already processed GOPs. The max and min functions are implemented to filter out the encoding bit parameters that result in $\bar{C}$-$\bar{Q}$ irregularities.

5. Simulation results and discussion

The performance of the proposed single-pass CBR and VBR video encoders is evaluated by compressing a library of test sequences. This library contains a diverse class of composite sequences, as described in **Table 1**. The video sequences are digitized according to the CCIR 601 resolution [11], with a color sampling ratio of 4:2:2. A 4:2:0 color sampling ratio is created by applying preprocessing filters on the chrominance samples of each source. The compression procedure is carried out in 4:2:0 mode using MPEG-2-compliant main profile at main level encoder software developed at our research laboratory. Since optimizing the communications bandwidth or the digital storage medium is very important for broadcast, consumer, and multimedia applications, most dedicated MPEG-2 encoders are configured to operate in the low-bit-rate regime of the main profile at the main level. Because of this, we assess the rate-distortion performance of the new algorithms at the popular rate of 4 Mb/s using the common peak signal-to-noise ratio (PSNR) measure defined by

$$PSNR = 10 \log_{10}\left(\frac{255^2}{D_{seq}}\right), \tag{44}$$

with

$$D_{seq} = \frac{1}{A_{seq}} \sum_{all(i,j,n)} (I^n_{i,j} - \hat{I}^n_{i,j})^2, \tag{45}$$

where $I^n_{i,j}$ and $\hat{I}^n_{i,j}$ are the intensities of the uncompressed picture n and its reconstructed version at position (i, j), respectively. The spatial indices i and j represent positions of rows and columns of a picture. Both luminance and chrominance image samples are included in the sum calculation of Equation (45). Finally, A_{seq} is the total number of pixels in a video sequence of 4:2:0 format.

Table 2 compares the average sequence PSNRs produced by all single-pass encoders against the results from the conventional approach of MPEG-2 compression denoted by "old CBR." The proposed single-pass CBR encoder is labeled "new CBR" throughout this section. Figures 9–12 illustrate the picture-bit usage for each GOP and the PSNR performance for some of the video sequences of Table 2. Test sequences mpeg and mixb are composed primarily of stressful video materials with high spatial details such as "Mobile and Calendar" and "Flower Garden" or medium to high amounts of motion, as in video shots of action scenes taken by a car camera. For the case of "Cheerleaders," used as part of the mpeg sequence, spatial and temporal activities are both significantly present. The efficiency of the VBR-1 and VBR-2 encoders depends on the accuracy with which the input stream can be classified into "hard" and "soft" GOPs. The perceptual $\bar{C}$-$\bar{Q}$ model can then take advantage of this classification and divert bits to video segments which are the most stressful. This observation can be deduced from the PSNR plot of the

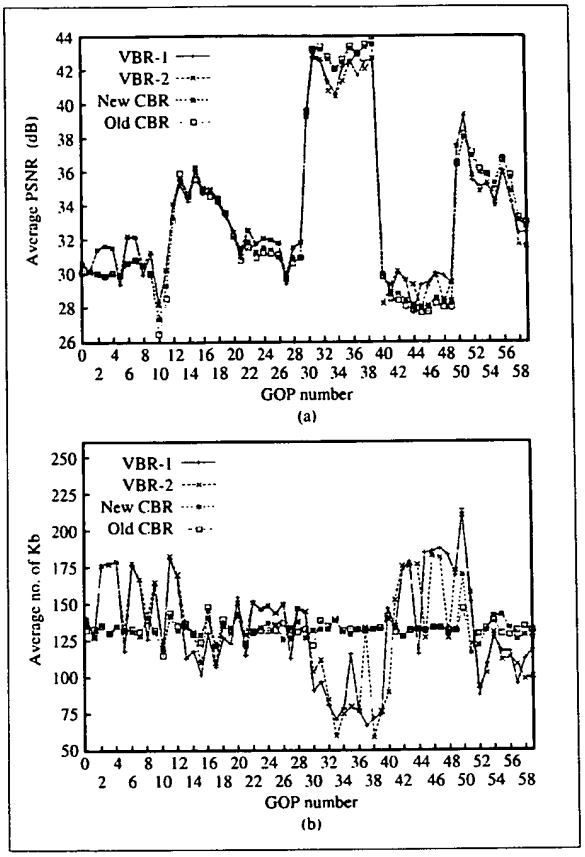

Figure 9

Compression results for the mpeg sequence at 4 Mb/s: (a) average PSNR per GOP; (b) average number of picture bits per GOP.

mpeg sequence in **Figure 9**. In this figure the GOP sets $\{S_t\}_{t=0}^{9}$, $\{S_t\}_{t=20}^{29}$, and $\{S_t\}_{t=40}^{49}$ represent "Cheerleaders," "Flower Garden," and "Mobile and Calendar" substreams, respectively, for which an increase of more than 1 dB in average PSNR is obtained. This quality improvement is achieved at the cost of some PSNR degradations in the rest of the sequence. However, the overall quality of the video is improved, as reflected by the numbers given in Table 2.

The new CBR encoder takes advantage of the increase in the local complexity of a picture to increase picture bits while allotting the same number of bits to each GOP. As a result, its PSNR performance lies somewhere between the single-pass VBR schemes and the old CBR. For "difficult-to-encode" pictures, the PSNR numbers are higher than those for old CBR and lower than for VBRs, while for the "easy-to-encode" pictures, the performance is higher than for VBRs and lower than for old CBR. The most difficult portion of the mixb sequence is the set defined by $\{S_t\}_{t=18}^{25}$, which contains scenes of buildings with many windows panning across the screen. For this video content, called "Skyscrapers," VBRs and new CBR offer better results than old CBR, as shown in **Figure 10**. The GreySep sequence displays an even better PSNR performance for its difficult video content when the new single-pass encoders are compared to the old CBR encoder. This is attributed to the nature of this source, which is composed of one-second natural scenes separated by one-second gray pictures. The gray pictures are created by setting the luminance and chrominance samples to a constant value. Since the gray segments require only a small number of bits to produce good-quality pictures, the VBR encoders can use the surplus bit budget to enhance the perceived quality of the non-gray pictures. As a result, the GreySep sequence, which is typically a very stressful test case for CBR encoders because of rapid scene changes, can easily be handled by the VBR encoder. For this example, improvements of 4 and 2 dB in average PSNR are noticed in **Figure 11** for the "Cheerleaders" and "Mobile and Calendar" subsequences, represented by $\{S_t\}_{t=26}^{27}$ and $\{S_t\}_{t=30}^{31}$, respectively.

Sequences SzBk and BkSz are composed of equal numbers of easy and difficult video segments, each five seconds long.

SzBk begins with "Susie" and ends with "Basketball," while BkSz has "Basketball" at the beginning and "Susie" as the trailer. An examination of PSNR figures (not shown here) revealed that these streams can easily be classified into different activity regions. The quality of the compressed streams is improved by allocating more bits to the "difficult-to-encode" "Basketball" subsequence at the cost of small degradations in the "easy-to-encode" "Susie" subsequence.

The LaBk sequence is a 20-second PAL source which can also be classified into two subsequences, with the exception that it contains a special-effect region. The first 10 seconds of the clip displays a person describing a diagram of a flowchart followed by a 10-second "Basketball" subsequence.[7] The flowchart undergoes special effects such as zoom, rotation, and spatially nonuniform intensity changes, which take place over several pictures. These forms of variation in image structures can seriously stress the efficiency of the motion-estimation technique and complicate the task of selecting accurate picture quantizers for any MPEG-2 encoder. On the contrary, our proposed single-pass video encoders can quickly adjust to such perturbations and provide better-quality streams. For the VBR encoder, we accomplish this by first examining the slope of the causal predictive model to detect anomalies in the compression parameters, and

[7] This Basketball clip is different from the one used in the SzBk and BkSz sequences.

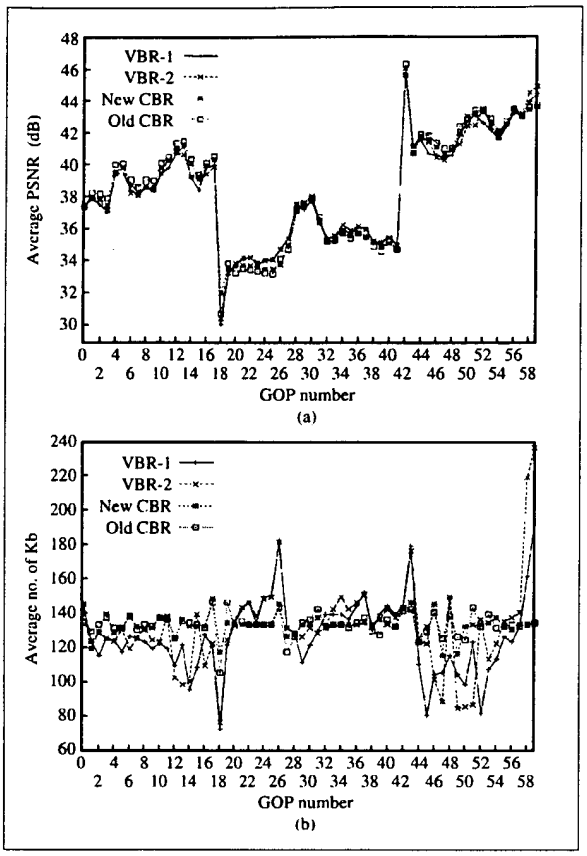

Figure 10

Compression results for the mixb sequence at 4 Mb/s: (a) average PSNR per GOP; (b) average number of picture bits per GOP.

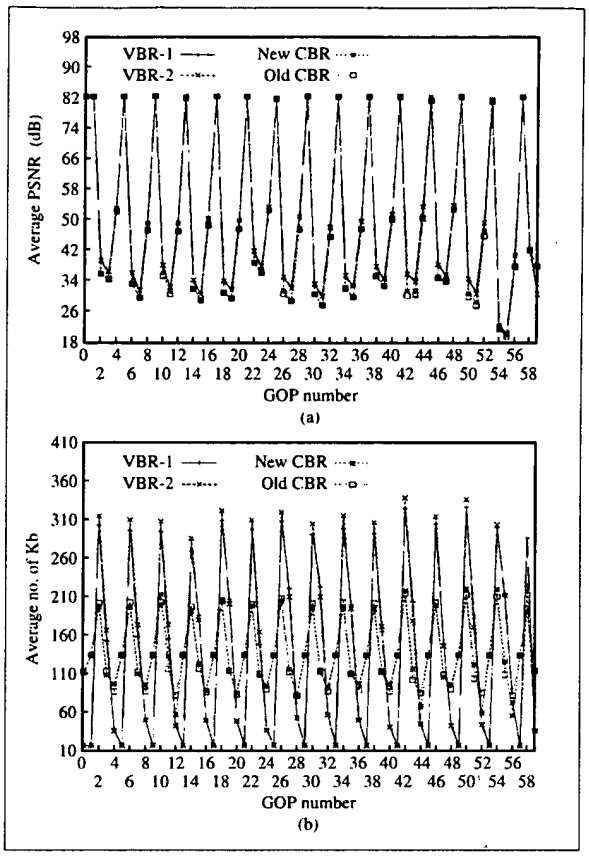

Figure 11

Compression results for the GreySep sequence at 4 Mb/s: (a) average PSNR per GOP; (b) average number of picture bits per GOP.

then setting the number of bits allocated to the next GOP using the rules defined in Equation (40) or (41). This results in better bit allocations for troublesome areas. The new CBR encoder improves the quality of the special-effect pictures by detecting an increase in the encoding difficulty (as a result of inaccuracies in the motion vectors or the picture quantization scaler) and enlarging the picture targets accordingly. **Figure 12** shows PSNR and bit usage of all encoders for the LaBk sequence. A comparison of the average PSNR numbers for the pictures in the special-effect region is given in **Table 3**.

Also presented in Table 3 are results for the Sprite sequence, which is composed of a segment where a fade to (from) black from (to) natural video material takes place. Fades are a special class of temporal image discontinuities which can seriously challenge the robustness of the motion-estimation technique of the encoder. As a result, B- and P-picture types do not do well during fades, and it is better to pre-analyze a collection of pictures, prior to encoding, and label them all-I if a fade is detected. Our single-pass encoders do not require a pre-analyzing step and are capable of adjusting their RC parameters in real time to compensate for the luminance changes of the pictures within the fade. Compression results for the Sprite sequence are given in Table 2. Finally, as the last test sequence, we encoded Singer, which comprises flashes of different intensities. For this case a small improvement in the PSNR performance is obtained, as indicated by the tabulated results of Table 2.

As previously stated in Section 4, the median filtering techniques of Equations (40) and (41) can be used to set the GOP target bits when a scene change is detected at the start of the GOP. This method is defined as GOP-level scene-change detection (GLSCD). We detect a scene change by comparing the weighted sum of differences of average intensities of consecutive reference pictures

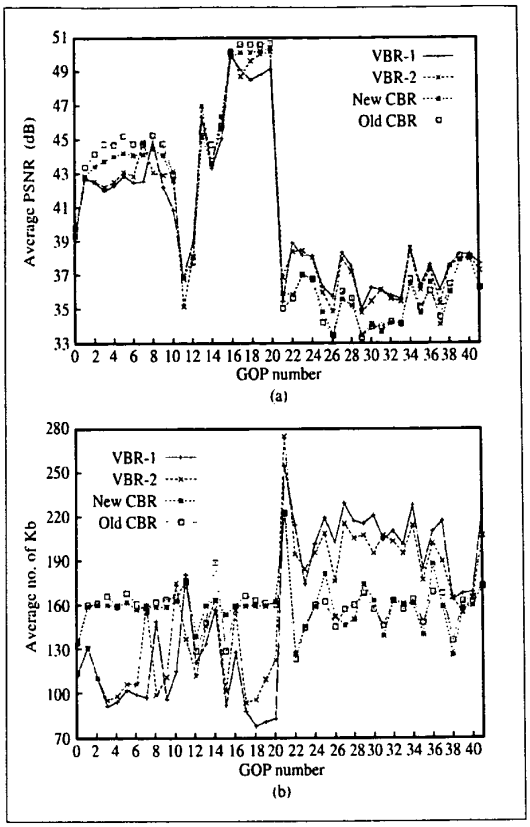

Figure 12

Compression results for the LaBk sequence at 4 Mb/s: (a) average PSNR per GOP; (b) average number of picture bits per GOP.

Table 3 Average PSNRs for special-effect regions of video streams encoded at 4 Mb/s.

Sequence	*Old CBR* (dB)	*New CBR* (dB)	*VBR-1* (dB)	*VBR-2* (dB)
Sprite	39.96	40.44	40.19	40.20
LaBk	39.42	39.52	40.12	39.10

against a predefined threshold. A reference picture can be either I- or P-type. Since a GOP always starts with an I-picture, and we have assumed a fixed GOP structure throughout this paper, intensity differences are computed between a current I-picture and a previous P-picture. Let Y_I^n represent the average intensity of the luminance component of an I-picture (in temporal position n) to be encoded at the start of the GOP, and let $Y_P^{n-(N^B+1)}$ represent the average intensity of the luminance component of a previous P-picture just encoded. The P-picture is displayed $(N^B + 1)$ pictures before the I-picture, since N^B is the number of B-type pictures. We further define similar representations for the average intensities of the chrominance components of the picture types under consideration by introducing statistical measures CB_I^n, $CB_P^{n-(N^B+1)}$, CR_I^n, and $CR_P^{n-(N^B+1)}$. A threshold τ_{th} is employed above which a scene change is declared using the decision

$$\frac{\Omega_1|Y_I^n - Y_P^{n-(N^B+1)}| + \Omega_2|CB_I^n - CB_P^{n-(N^B+1)}| + \Omega_3|CR_I^n - CR_P^{n-(N^B+1)}|}{\sum_{i=1}^{3}\Omega_i} > \tau_{th}. \quad (46)$$

The set of weights $\{\Omega_1, \Omega_2, \Omega_3\}$ is chosen so as to give importance to certain statistical measures. We label the VBR encoding scheme which adopts the scene-change criterion of Equation (46) as VBR w/GLSCD. The impact of the scene-change detection strategy on the quality of the VBR streams is assessed by compressing video sequences mpeg, mixb, and GreySep. A comparison of the PSNR performance of the VBR encoders with and without the GLSCD formulation is provided in **Table 4**. The importance of the GLSCD addition is not reflected by the PSNR improvements of Table 4. An examination of the PSNR figures (not shown here) revealed that detecting video discontinuities and applying the median filtering of (40) and (41) cleans up some of the undesired bit targets that are allocated between the scenes of different complexities.

We also simulated the highly detailed video sequences mpeg and mixb using the GOP bit-irregularity detection (GBID) methodology defined by (42). This approach improves the quality of the difficult segments of sources such as "Cheerleaders" and "Mobile and Calendar" of the mpeg sequence and "Skyscrapers" of the mixb sequence, as shown by the PSNR plots of **Figures 13** and **14**. In these regions, GOP bit irregularities are filtered, and a more intelligent bit distribution is achieved. **Table 5** summarizes average-sequence PSNRs obtained by using the VBR encoders with and without incorporating the GLSCD and GBID methods.

The rate-distortion analysis of the VBR formulations suggests that the VBR-2 encoder allocates more bits to the beginning of the sequence in comparison with the VBR-1 encoder. This is because in the VBR-2 scheme the perceptual model is translated after each GOP is encoded. This provides a faster reaction to compensate for the underproduction of bits if the sequence is started with "soft" GOPs. For the case in which the source starts with a "hard" GOP, this reaction takes longer to develop. On the basis of this observation and the results presented in this paper, the VBR-1 encoder produces better video streams for sources which are composed of many

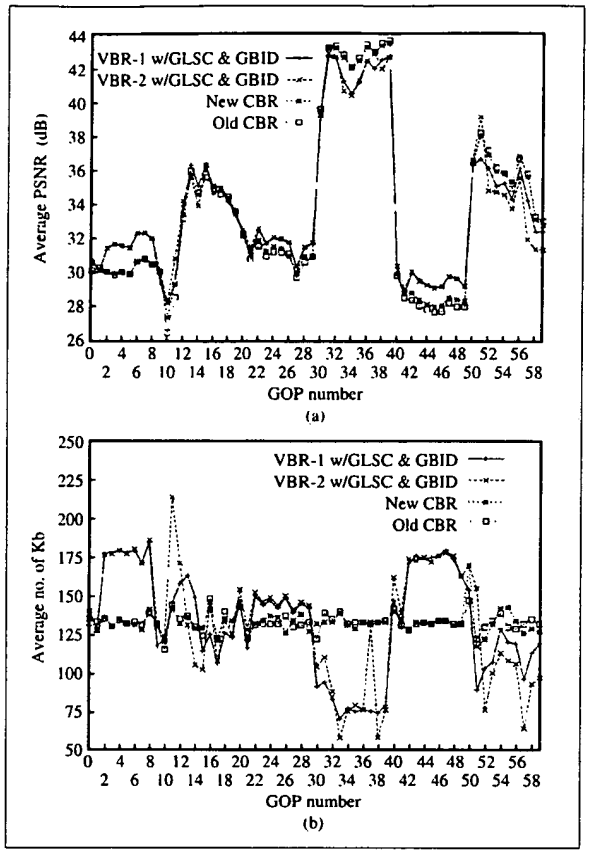

Figure 13

Compression results for the mpeg sequence at 4 Mb/s with GLSCD and GBID enabled: (a) average PSNR per GOP; (b) average number of picture bits per GOP.

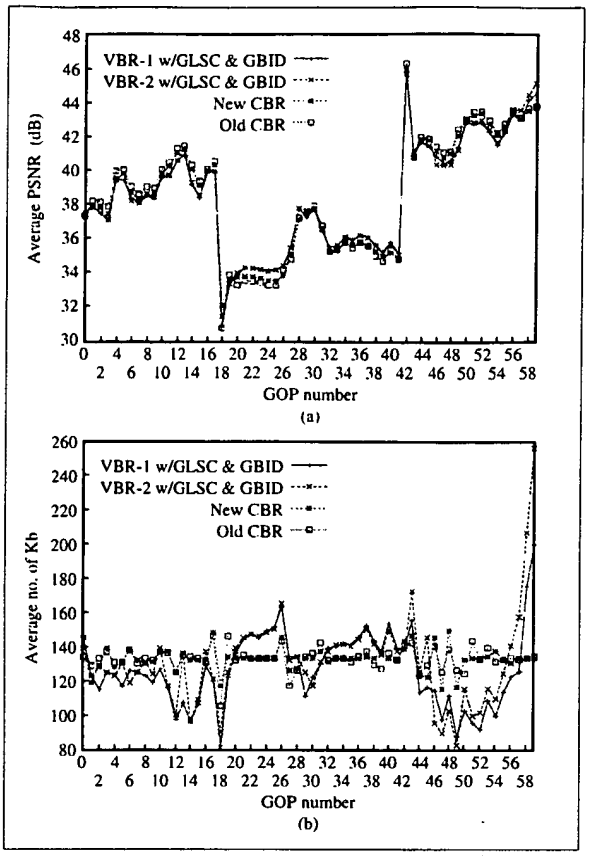

Figure 14

Compression results for the mixb sequence at 4 Mb/s with GLSCD and GBID enabled: (a) average PSNR per GOP; (b) average number of picture bits per GOP.

Table 4 Average PSNRs and actual bit rates for a few test sequences with the GOP-level scene-change detection (GLSCD) enabled and disabled.

Sequence	*VBR-1*	*VBR-1 w/GLSCD*	*VBR-2*	*VBR-2 w/GLSCD*
mpeg	32.14 dB @ 4.0 Mb/s	32.16 dB @ 3.99 Mb/s	32.10 dB @ 4.04 Mb/s	32.16 dB @ 4.03 Mb/s
mixb	37.02 dB @ 3.8 Mb/s	37.08 dB @ 3.79 Mb/s	37.10 dB @ 3.96 Mb/s	37.15 dB @ 3.96 Mb/s
GreySep	32.65 dB @ 4.02 Mb/s	32.69 dB @ 3.97 Mb/s	32.66 dB @ 4.07 Mb/s	32.57 dB @ 4.0 Mb/s

Table 5 Comparison of average PSNRs for highly detailed video sequences with GOP-level scene-change detection (GLSCD) and GOP bit-irregularity detection (GBID) techniques enabled and disabled. Actual bit rates are also given.

Sequence	*VBR-1*	*VBR-1 w/GLSCD & GBID*	*VBR-2*	*VBR-2 w/GLSCD & GBID*
mpeg	32.14 dB @ 4.0 Mb/s	32.21 dB @ 4.0 Mb/s	32.10 dB @ 4.04 Mb/s	32.22 dB @ 4.05 Mb/s
mixb	37.02 dB @ 3.8 Mb/s	37.10 dB @ 3.8 Mb/s	37.10 dB @ 3.96 Mb/s	37.17 dB @ 3.96 Mb/s

dissimilar GOPs in terms of grades of "hardness" and "softness." On the other hand, the VBR-2 encoder is very useful for compression of sources which are biased toward "soft" GOPs.

6. Conclusion

In this paper, we propose new formulations for single-pass constant-bit-rate and variable-bit-rate encoding schemes for an MPEG-2-like environment. The real-time implementations of these schemes are easily accomplished with IBM encoder architecture. We have found that differentiating the local complexity of a picture from that of the rest of the encoded pictures can be used as a difficulty measure to increase the perceived quality of difficult pictures of the video while still adhering to the constant bit rate of the compressed bitstream. In comparison with the conventional approaches of CBR encoding, our single-pass CBR encoder shows better performance. This performance can further be improved by creating a VBR bitstream. We have taken advantage of both statistical and perceptual rate-quantization models to estimate the variability of the video sequence, thereby economizing the available bit budget in a single-pass encoding scenario. This is achieved by not wasting bits in the simple video segments while allocating more bits to areas where the penalty for quality degradation is greater. Our single-pass VBR strategy presents a more consistent perceptual quality throughout the stream than do the CBR encoders.

The VBR framework of this paper can be improved upon to accommodate the requirements of the next generation of digital storage media capable of storing high-definition video sequences. We can accomplish this by establishing more sophisticated rate-quantization profiles to better predict the allotted GOP bits. Ongoing advances in VLSI and fabrication technologies should provide us with the necessary architecture to achieve this goal in the future with a single-chip encoder dedicated to real-time applications.

References

1. "Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video," First Edition, *ISO/IEC 13818-2*, May 1996.
2. "Test Model 5," *ISO/IEC JTC1/SC29/WG11/N0400*, April 1993.
3. G. Ramamurthy, D. Raychaudhuri, and D. J. Reininger, "VBR MPEG Video Encoding for ATM Networks with Dynamic Bandwidth Renegotiation," U.S. Patent 5,675,384, October 7, 1997.
4. P. Pancha and M. El Zarki, "Bandwidth-Allocation Schemes for Variable-Bit-Rate MPEG Sources in ATM Networks," *IEEE Trans. Circuits & Syst. for Video Technol.* **3**, No. 3, 190–198 (June 1993).
5. M. R. Pickering and J. F. Arnold, "A Perceptually Efficient VBR Rate Control Algorithm," *IEEE Trans. Image Process.* **3**, No. 5, 527–532 (September 1994).
6. J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*, Chapman and Hall Publishers, New York, 1997.
7. E. Linzer, "A Robust MPEG-2 Rate Control Algorithm," *Research Report RC-20361*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, March 1996.
8. A. R. Reibman and B. G. Haskell, "Constraints on Variable Bit-Rate Video for ATM Networks," *IEEE Trans. Circuits & Syst. for Video Technol.* **2**, No. 4, 361–372 (December 1992).
9. P. H. Westerink, R. Rajagopalan, and C. A. Gonzales, "Two-Pass MPEG-2 Variable-Bit-Rate Encoding," *IBM J. Res. Develop.* **43**, No. 4, 471–488 (1999, this issue).
10. A. J. Seyler and Z. L. Budrikis, "Detail Perception After Scene Changes in Television Image Presentations," *IEEE Trans. Info. Theory* **IT-11**, 31–43 (January 1965).
11. "Encoding Parameters of Digital Television for Studios," *CCIR Recommendation 601*, in *CCIR Recommendation and Reports*, Vol. XI, International Telecommunications Union, Geneva, Switzerland, 1982.

Received May 18, 1998; accepted for publication June 17, 1999

Nader Mohsenian *IBM Research Division, 1701 North Street, Endicott, New York 13760 (nmohseni@us.ibm.com).*
Dr. Mohsenian graduated from Lehigh University with B.S. and M.S. degrees in electrical engineering. He received his Ph.D. in electrical engineering from Worcester Polytechnic Institute in 1992. From 1992 to 1994 he was a Postdoctoral Research Fellow with the Department of Electrical Engineering of Princeton University. In 1994 he joined the research and development staff of the IBM Digital Video Products group, where he is currently a Senior Scientist. His work is focused on developing digital video processing techniques for MPEG-2 products. Dr. Mohsenian has published several papers on digital image and video compression. He is a member of IEEE, SMPTE, and Eta Kappa Nu.

Rajesh Rajagopalan *Lucent Technologies, 1160 Route 22E, Room 227B, Bridgewater, New Jersey 08807.* Dr. Rajagopalan received the Bachelor of Engineering degree in electronics and communication engineering in 1986 from the Regional Engineering College, Suratkal, India, and the Master of Science degree from the University of Texas at Austin in 1993 and the Ph.D. degree from the University of Illinois at Urbana-Champaign in 1996, both in electrical and computer engineering. He has been a Research Assistant in the Electrical Engineering departments of the University of Texas at Austin and the University of Illinois at Urbana-Champaign, a Research Associate at the Institute for Geophysics, Austin, Texas, a visiting student at Princeton University, and a Research Staff Member at the IBM Thomas J. Watson Research Center at various times from 1990 through 1997, working in the areas of signal, image, and video processing. He is currently a Member of Technical Staff at Lucent Technologies, working in the field of digital video. His areas of interest include video processing, distribution, and communications.

Cesar A. Gonzales *IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598 (butron@us.ibm.com).*
Dr. Gonzales is an IBM Fellow, Senior Manager for Multimedia Technologies at the IBM Thomas J. Watson Research Center, and manager of the development organization of the IBM Digital Video Products Group. He is an expert in image and video processing and compression; his experience spans the development of algorithms, chip and system architectures, and multimedia applications. He is a co-inventor of various still-frame and motion video compression techniques that IBM contributed to the JPEG and MPEG international standards. He is the author of number of patents and publications related to these techniques. While at IBM, he has received multiple Outstanding Achievement awards, including a Corporate-level award for his contributions to IBM's MPEG products. In June 1998 Dr. Gonzales was named an IBM Fellow. He was also elected to serve in IBM's Academy of Technology, and has twice been named Master Inventor for his contributions to IBM's patent portfolio. Prior to coming to IBM, Dr. Gonzales worked on radar signal processing and physics of the ionosphere at the Arecibo Observatory in Puerto Rico. Dr. Gonzales is a Senior Member of the IEEE. He has served as an Associate Editor of the *IEEE Transactions for Circuits and Systems for Video Technology*. He has also served as the head of the U.S. delegation in the MPEG committee of the International Standards Organization. Dr. Gonzales received his B.S. and engineering degrees from the National University of Engineering (UNI) in Peru and his Ph.D. from Cornell University, all in electrical engineering.